

情報(1) H8 マイコンによる A/D 変換処理及び LED での結果表示

1. 目的

温度、気圧、電気抵抗、電圧など、私たちの身の回りに様々な物理量がある。それらを計測するために、いろいろな計測器が開発され、使われている。多くの計測器はマイコン、センサ（入力）、表示デバイス（出力）により構成されている。これらの計測機器の構成と制御を理解するために、本実験は H8 マイコンを用いて、ポテンシオメータ（電気抵抗値）を測定し、LED で表示する方法を学ぶことを目的とする。

2. 実験

この実験では、e-nuvo BASIC のベース基板上にあるポテンシオメータ（ボリューム）と LED を用いて、CPU の A/D 変換機能について学習する。具体的に、ベース基板右下にあるポテンシオメータの電気抵抗値より変換された電圧（アナログ値）を H8CPU に入力し、A/D 変換処理を行った結果（デジタル値）を LED（8 ビット）に出力する。図 1 は本実験の入出力構成を示している。

なお、本実験の操作は、「電子情報実験 I」で行った「H8 マイコンを用いた電子情報システム実験入門」などで習った実験手順とほぼ同じであり、実験を行う前に、必ず「電磁情報実験 I」の課題 10、課題 11 及び課題 12 の実験マニュアル（以下参照資料という）を復習すること。自信がない人は、これらの資料も必ず持参すること。

3. 実験手順

3.1 必要機材

本実験では、「電子情報実験 I」のマイコン関連実験で用いた機材と同じものを使用する。プログラムの編集、コンパイルおよびフラッシュメモリへの書き込み等も同じツールを利用する。主な必要機材は下記に示す。なお、これらの機材はすべて実験台に揃えられている。

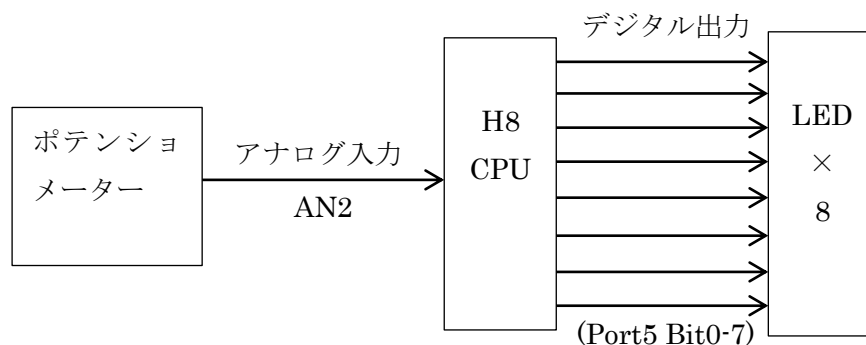


図 1 本実験の入出力構成

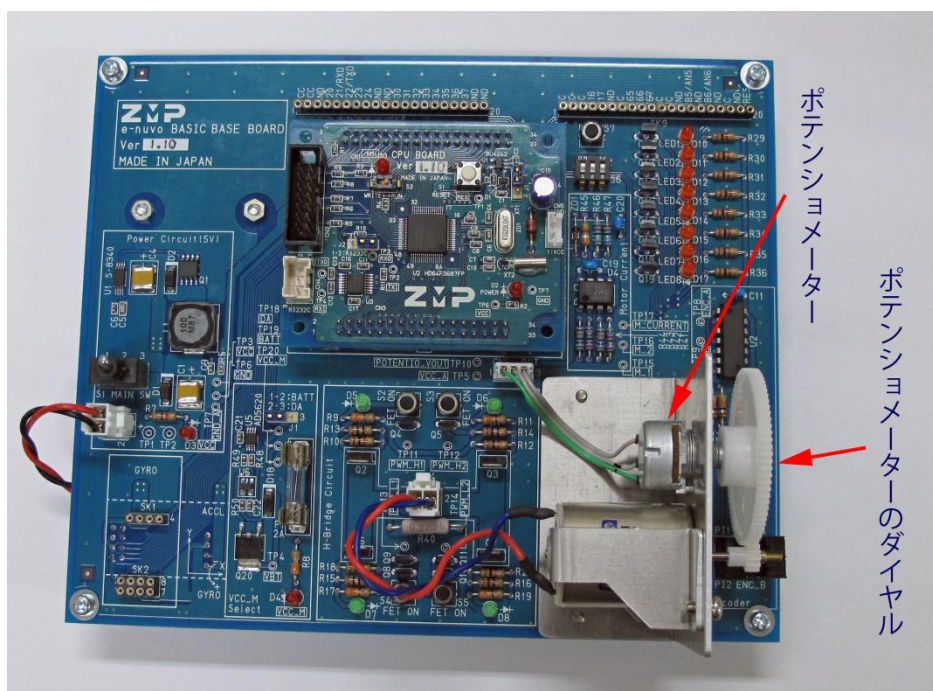


図2 e-nuvo BASIC (BASE BOARD & CPU BOARD)

- e-nuvo BASIC (図2)
- 各種ケーブル、ACアダプター、
- パソコン及び開発環境一式

3.2 実験の手順

実験の流れと手順を以下に示す。

A) 新規プロジェクトの作成

H8の統合開発環境であるHEWを起動し、新規プロジェクトを作成し、プログラミングの準備をする。各種パラメータを下記のように設定する。

- ・ ワークスペース： AD_CONV_LED
- ・ CPU 種別： 300H
- ・ CPU タイプ： 3687
- ・ ディレクトリ： z:\¥AD_CONV_LED

また、プロジェクトに、ヘッダファイル「3687.h」を追加する。

(不明な場合、関連資料を参照すること。)

B) プログラムの作成

C言語で、A/D変換、LEDの点灯プログラムを作成する。具体的に、HEWの中で、AD_CONV_LED.cを下記のように書き換える(網掛けされた箇所は、修正部分である)。網掛けされた部分以外も下記リストと異なる部分もあるかも知れないが、気になくて良い。基本は、main関数の中の書き換えのみである。

```

#include "3687.h"

void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

void main(void)
{
    //初期環境設定
    IO.PCR5 = 0xFF;      // LED1~8 を出力とする
    // A/D 変換設定
    AD.ADCSR.BIT.ADST = 0;    // A/D 変換を停止
    AD.ADCSR.BIT.SCAN = 0;    // A/D 変換のモード設定
    AD.ADCSR.BIT.CKS  = 1;    // 変換時間を設定
    AD.ADCSR.BIT.CH   = 2;    // アナログ入力チャンネルの選択

    while (1)
    {
        // A/D 変換を開始
        AD.ADCSR.BIT.ADST = 1;

        // A/D 変換終了まで待機
        while (AD.ADCSR.BIT.ADF==0) {}

        // A/D 変換結果を LED 出力
        IO.PDR5.BYTE = (AD.ADDRC>>8);
    }
}

#ifdef __cplusplus
void abort(void)
{
}
#endif

```

C) コンパイル

作成したプログラムをコンパイルし、H8 マイコンが理解できる形式に変換する。
(不明な場合、関連資料を参照すること。)

D) プログラムをマイコンに書き込む

PC と H8 マイコンを接続し、コンパイルしたプログラムをマイコンに書き込む。
(不明な場合、関連資料を参照すること。)

E) 動作確認

電源を入れ、プログラム通り、ポテンショメーターのダイヤルを回して、LED の点灯がそれに従って変化することを確認する。ポテンショメーターの抵抗値を大きく又は小さくすると、LED の点灯で表示する値（2 進数）も大きく又は小さくなるのであれば、実験が成功である。

4. 課題

・課題 1

- (1) 作成したプログラムのフローチャートを描き、説明せよ。
- (2) A/D 変換結果を LED 出力する文「IO.PDR5.BYTE = (AD.ADDRC>>8);」について、実際何が行われたか説明すること。

ヒント：

A/D 変換した結果が格納されるのは ADDRC であるが、ADDRC は 16 ビットのレジスタであるのに対して、A/D 変換結果は 10 ビットの値である。実際は、ADDRC の MSB（最上位ビット）から格納していくことになっている。それを LED 出力（PDR5）の 8 ビットにマッチさせるためにはどうすればよいかを説明すればよい。図 3 はそのマッチングの模式図である。

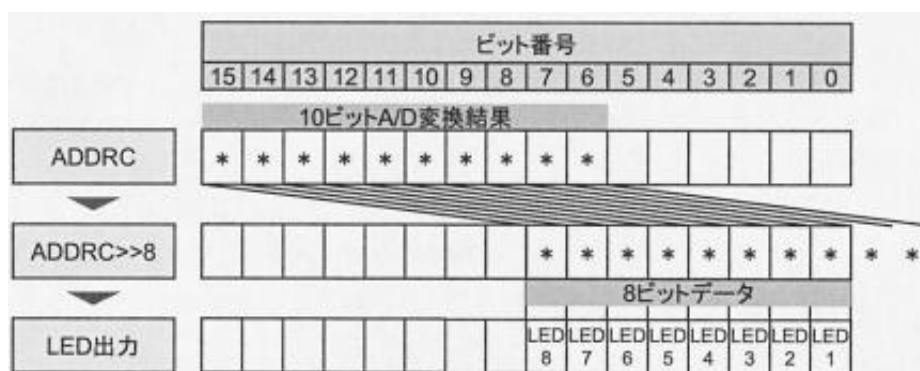


図 3 A/D 変換結果と LED 出力との関係

・課題 2

上記の実験で、ポテンショメーターの抵抗値を LED で表示することができたが、抵抗値の大きさと LED 表示との関係は必ずしも直感的ではない。オーディオ機器のボリューム等のデザインによく見られるように、抵抗値の大きさに応じて、LED 点灯の個数を変化させるようにする。例えば、ある抵抗値に対応する LED 点灯状態が

○○○○○●●●●

になっている（●は点灯を表す）とすると、抵抗値がある程度増えると、次の点灯状態が、

○○○○○●●●●●

となるようにする。最終的に、抵抗値が最大となったときに、LED もすべて点灯する（すべて●状態）。レポートに、その改良に当たる、AD_CONV_LED.c の中の修正部分を書き出せ。また実験結果について考察せよ。

（ヒント：点灯パターンは 9 通りあるので、A/D 変換の結果がその 9 通りのどのパターンになるかを判断して、LED の出力を設定すればよい。）

- ・ 課題 3

課題 2 の改良プログラムをさらに改善するために、ポテンシオメーターを回す角度と点灯状態の遷移との関係の調整し、プログラムに反映する。例えば、ポテンシオメーターの回転角度が 360 度だとすると、45 度を回す度に、LED が 1 個点灯するまたは 1 個消灯するようにする。その改良に当たる、AD_CONV_LED.c の中の修正部分を書き出せ。また実験結果について考察せよ。